Vol 5 (1) (2024)

Measurement, Control, and Automation

Website: https:// mca-journal.org

ISSN 1859-0551

A Comparative Study of Machine Learning – based Models for Short-Term Multi-step Forecasting of Solar Power: An Application for Nhi Ha Solar Farm

Thu Nguyen Thi Hoai [1,*], Van Pham Nang [1], Khanh Ngo Van [1], Bach Do Xuan [1]

[1] *Power Grid and Renewable Energy Lab., School of Electrical and Electronic Engineering, Hanoi University of Science and Technology, Hanoi, Vietnam*

[*] *Corresponding author E-mail: thu.nguyenthihoai@hust.edu.vn*

Abstract

Over the past few decades, the utilization of solar power has gained immense significance in the power grid, gradually taking over the responsibilities of fossil fuel-based power. Therefore, accurate short-term forecasting of photovoltaic power output is crucial for making informed decisions regarding power generation, transmission, and distribution. Consequently, many machine-learning models were used to reliably forecast solar power. In this study, four machine learning models have been studied which are Artificial Neural Networks, Convolutional Neural Networks, Long Short-Term Memory (LSTM) and Extreme Learning Machine (ELM). They have been used to forecast the solar power of Nhi Ha solar farm in short-term. First, data from Nhi Ha solar farm were collected and underwent preprocessing before being utilized by aforementioned distinct machine learning models. The Root Mean Squared Error (RMSE) and normalized RMSE (N-RMSE) obtained from the models will be analyzed to determine the most effective model for short-term solar power forecasting. Following a comprehensive analysis, it has been determined that all four models have produced favorable outcomes, with low values of RMSE and N-RMSE indicating high levels of reliability and accuracy. Of the models considered, the LSTM and ELM models have demonstrated better performance, making them the good choice for precise short-term solar power forecasting.

Keywords*: Artificial Neural Networks, Convolutional Neural Networks, Machine learning, Short-term forecasting, Solar power*

Abbreviations

ANN	Artificial Neural Networks
CNN	Convolutional Neural Networks
ELM	Extreme learning machine
LSTM	Long short-term memory
RMSE	Root Mean Squared Error
N-RMSE	Normalized Root Mean Squared Error

1. Introduction

In recent years, the utilization of solar power in Vietnam has experienced a remarkable surge, with the government implementing plans to promote further use of this renewable energy source. In 2020, Decision 13/2020/QD-TTg was enacted, which introduced a variety of new incentives and mechanisms aimed at attracting investors to solar power projects within the country [1]. Considering Vietnam's accelerated expansion in the renewable energy sector, particularly in electricity, solar power is progressively becoming an indispensable element of Vietnam's power grid. This necessitates the development of precise and accurate forecasting models. The integration of solar energy into the power grid presents numerous challenges due to the highly volatile and unpredictable nature of solar energy sources. These characteristics are influenced by various environmental factors such as sunlight intensity, wind speed, and temperature. Any inaccuracies in forecasting can lead to errors in power grid management, potentially resulting in significant social and economic impacts that could disrupt businesses and civilian life. Consequently, the accurate forecasting of solar power is essential for ensuring the stability and maintenance of the power grid. The evolution and application of various forecasting models and technologies over recent decades have simplified the management of electricity in Vietnam, facilitating a more stable and seamless integration of solar power into the power grid.

Over the past several decades, numerous techniques have been developed to forecast the capacity of solar farms[2]. These forecasting methods can be classified in several ways, one of which is based on the forecasting models themselves. These categories include: Statistical methods (time series-based methods); Artificial intelligence methods; Physical methods; and Mixed methods (hybrid or ensemble methods) [3]. Statistical methods such as Autoregressive Moving Average (ARMA) [4], [5] or Autoregressive Integrated Moving Average (ARIMA) [6] have been widely used in various short-term solar power forecasting applications. However, these models often encounter several limitations, such as their inability to accurately forecast non-linear data or their dependence on historical data. To mitigate these inaccuracies, machine learning models or machine learning models are employed, yielding improved results.

Received: 21 November 2023; Accepted: 17 January 2024.

Owing to their superior performance, machine learning models have been increasingly adopted for short-term solar power forecasting. Common machine learning models include Artificial Neural Network (ANN), Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Auto-encoder (AE), among others. ANN is one of the most effective methods due to its adaptability to large fluctuations caused by changing environmental conditions [7]. CNNs have demonstrated significant effectiveness in short-term solar power forecasting. Utilizing temperature and solar irradiation as predictors, this algorithm has been used to predict the power output of a Photovoltaic (PV) system located in Italy, yielding satisfactory prediction results with a 10% error value. Lastly, LSTM is a model developed from the RNN model with the aim of mitigating RNN's weaknesses. LSTM-based model is particularly effective in dealing with time-series data and has demonstrated superior overall prediction accuracy for short-term forecasting [8] [9].

The primary contributions of this paper can be articulated as follows:

1. This paper studies four machine learning algorithms: Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM) and Extreme Learning Machine (ELM). These algorithms are configured appropriately, and their results are compared to ascertain which algorithm yields the most accurate results. The comparison of these four algorithms could inform enhancements to future forecasting models, thereby assisting operators in maintaining power system reliability, preserving power quality, and mitigating uncertainty in the grid.
2. This paper aims to understand and apply 4 mentioned machine learning - based forecasting models to data collected from the Nhi Ha solar farm.

2. Methodology

2.1. Artificial Neural Network

Artificial Neural Network (ANN) is a type of machine learning model that mimics the human brain in information processing, problem-solving and self-learning abilities [10]. The ANN model presents several advantages, including its capacity to operate effectively with incomplete datasets and the ability to retain information throughout the entire network rather than relying on a database [11]. Nevertheless, it is crucial to acknowledge that the ANN is susceptible to overfitting and underfitting phenomena, exerting an impact on both the performance and generalization capabilities of the network [11]. ANN is composed of several parts mainly: Input layer, Hidden layer, and Output layer. The input layer is the first layer of nodes in ANN with the mission to receive input data from external sources. The hidden layers then take the information for processing. The obtained value from the hidden layers is sent to the output for further processing to obtain the final result. The output signal can be expressed as:

$$P_j = \sum_{k=0}^{n} w_{jk} h_k \quad (1)$$

Where:

P_j: Represent the output signal

w_{jk}: Represent the weighted strength of the connection between the neuron (j) and the hidden layer (k)

h_k: Represent the weighted sum of n signals for each hidden layer of k=1,2,3,…,n

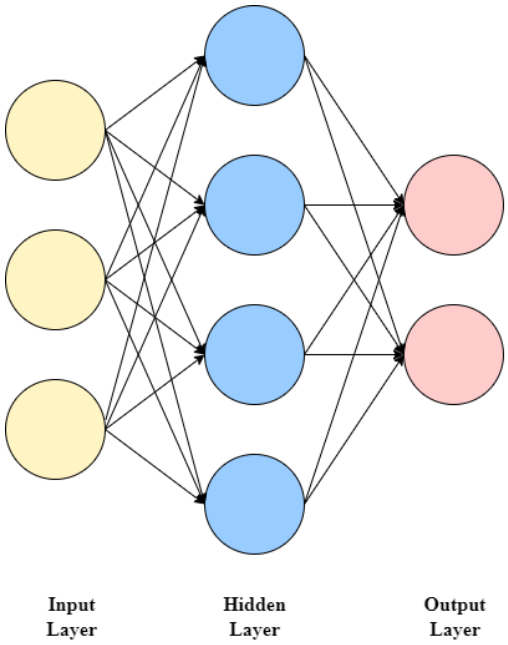

Figure 1: The structure of an ANN

2.2. Convolutional Neural Network

CNN is a type of feed-forward neural network mainly built for pattern recognition in images. It is mostly used for Object Detection [12], Image Classification [13], Face Recognition [14] etc. CNN is utilized in Computer Vision problems where data is composed of images passing through some optimization function and three layers namely: Convolutional Layer, Pooling Layer and Fully connected layer.

Apart from solving pattern recognition in image and Computer Vision problems, CNN can also be utilized in time-series forecasting by changing the dimensional structure of the input information.

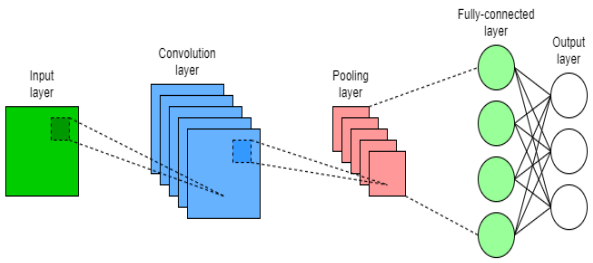

Figure 2: The structure of a CNN model

The following equation shows the output O of CNN calculation of some elements, where the Convolutional kernel is matrix P.

$$\begin{Bmatrix} o_{11} = f(\Sigma J \otimes P + b_1) \\ o_{21} = f(\Sigma J \otimes P + b_1) \\ o_{31} = f(\Sigma J \otimes P + b_1) \\ o_{12} = f(\Sigma J \otimes P + b_2) \\ o_{22} = f(\Sigma J \otimes P + b_2) \\ o_{32} = f(\Sigma J \otimes P + b_2) \\ o_{63} = f(\Sigma J \otimes P + b_3) \end{Bmatrix} \quad (2)$$

Beyond its advantages, the CNN model necessitates a substantial volume of data and resources for the training and optimization of network parameters. Furthermore, specialized hardware is imperative to expedite the computational processes associated with CNN, underscoring an additional requirement in its deployment. [15]

2.3. Long short-term memory

Recurrent Neural Networks (RNNs) are powerful machine learning models that have found use in a wide range of areas such as Speech Recognition [16], language translation [17] and image captioning problems [18]. When training a RNN model, one of the biggest problems is the "vanishing gradient". Long short-term memory (LSTM) [19] is a kind of RNN that was introduced to deal with the "vanishing gradient" problem in conventional RNNs. Furthermore, the LSTM model exhibits the capability to capture long-range dependencies and retain information over prolonged durations, rendering it well-suited for applications such as natural language processing and time series analysis [17]. Nevertheless, it is noteworthy that LSTM entails lengthier training periods and demands greater memory resources compared to simpler models, posing challenges in the context of large-scale applications [17]. Additionally, the susceptibility of LSTM to overfitting is particularly pronounced when confronted with limited or noisy training data. While dropout serves as a conventional technique to mitigate overfitting, its implementation in LSTM is more intricate compared to feedforward networks [17]. A LSTM unit is composed of: Forget gates, Input gates, the cell and the output gates. The forget gates filter redundant information by multiplying the value in the memory cell by a number of 0 or 1. The input gates determine which of the input values should be used to change the memory. The cell stores important information and the output gates control which information in the cell is kept as input value in the next timestep.

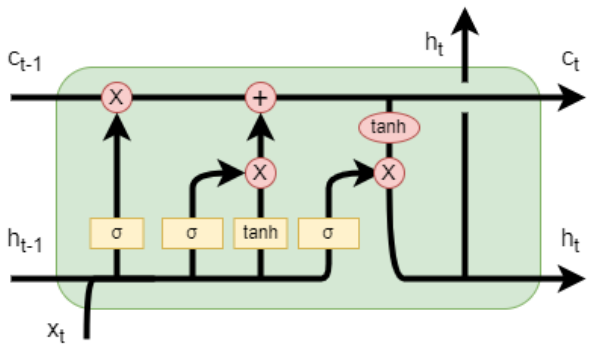

Figure 3: The structure of a LSTM unit

A LSTM block's mathematical model is presented by as follows:

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i) \quad (3)$$
$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f) \quad (4)$$
$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o) \quad (5)$$
$$\bar{c}_t = tanh(w_c[h_{t-1}, x_t] + b_c) \quad (6)$$
$$c_t = (f \otimes c_{t-1}) \oplus (i_t \otimes \bar{c}_t) \quad (7)$$
$$h_t = o_t * \tanh(c_t) \quad (8)$$

Where:

i_t: represents input gate
f_t: represents forget gate
o_t: represents output gate
i_t: represents input gate
σ: represents sigmoid activation function
w_x: weight for the respective gate(x) neurons
h_{t-1}: output of the previous lstm block (at timestamp t-1)
x_t: input at current timestamp
b_x: biases for the respective gates(x)
c_t: cell state at timestamp(t)
$\bar{c}_t$: represents candidate for cell state at timestamp(t)

2.4. Extreme learning machine

ELM is a learning algorithm invented for the purpose of training single hidden layer feedforward neural network (SLFNs) and is known for its fast convergence and promising performance [20]. In contrast to the iterative interlayer weight updates in traditional neural network training algorithms, which utilize backpropagation, the Extreme Learning Machine (ELM) adopts a different approach. The weights connecting the input layer and the hidden layer are established through random initialization. Subsequently, the weights between the hidden layer and the output layer are learned using the least squares method. This methodology results in a rapid training process during the learning stage and swift inference during the testing stage. However, ELM's randomly assigned hidden layer parameters can also lead to overfitting, especially when the number of hidden nodes is large compared to the training data size [20]. While ELMs offer faster training compared to traditional neural networks, they also have less flexibility in terms of fine-tuning the model architecture. There is less control over the specific connections and weights in the hidden layer, which can limit the model's ability to capture complex relationships in the data [20]. Some applications of the model include classification [21] and regression [20] tasks.

For N different training data pairs (X_i , Y_i), $X_i = \left|x_{i_1}, x_{i_2}, \ldots, x_{ik}\right|^{\mathrm{T}} \in R^k$ is the ELM model input and $Y_i = \left| y_{i_1}, y_{i_2}, \ldots, y_{\mathrm{im}} \right|^{f' \in R^m}$ is the expected model output. If the activation function of the ELM with L hidden layer nodes is $g(x)$, the output function expression of the ELM is shown in Eq. (9).

$$\sum_{l=1}^{L} \beta_l g(b_l + W_l \cdot X_i) = O_i, i = 1, \ldots, N \quad (9)$$

where b_l is the threshold of the lth node of the hidden layer, $W_l = \left[\omega_{l_1}, \omega_{l_2}, \ldots, \omega_{\mathrm{lk}}\right]^{\mathrm{T}}$ is the weight connecting the input layer node and the lth hidden layer node, $\beta_{\mathrm{l}} = \left[\beta_{1_1}, \beta_{l_2}, \ldots, \beta_{\mathrm{lm}}\right]^{\mathrm{T}}$ is the weight connecting the lth hidden

layer node and the output layer node, $W_l \cdot X_i$ illustrates the inner product of W_l and X_i, and O_i is the output of the EIM model. In the ELM model calculation, the model output and expected output are equal; therefore, Eq. (10) can be obtained as follows:

$$\sum_{l=1}^{L} \beta_l g(b_1 + W_l \cdot X_i) = Y_i, i = 1, \dots, N \quad (10)$$

Eq. (10) is converted into the matrix representation in Eq. (11).

$$H\beta = Y \quad (11)$$

Where:

$$H = \begin{bmatrix} g(X_1 \cdot W_1 + b_1) & \cdots & g(X_1 \cdot W_L + b_L) \\ \vdots & \ddots & \vdots \\ g(X_N \cdot W_1 + b_1) & \cdots & g(X_N \cdot W_L + b_L) \end{bmatrix}_{N \times L} \quad (12)$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m} \quad Y = \begin{bmatrix} Y_1^T \\ \vdots \\ Y_N^T \end{bmatrix}_{N \times m} \quad (13)$$

In the ELM model parameter training process, if $\bar{W}_l, \bar{b}_l$, and $\bar{\beta}_l$ can make Eq. (14) hold.

$$E = \min_{W,b,\beta} \sum_{L=1}^{N} \left(\sum_{L=1}^{L} \bar{\beta}_l g\left(\bar{W}_l \cdot X_l + \bar{b}_l\right) - Y_i \right)^2 = \min_{W,b,\beta} \| H\beta - Y \| \quad (14)$$

Then $\bar{W}_l, \bar{b}_l$ and $\bar{\beta}_l$ are the optimal ELM model parameters. In the ELM model, the determination of the input weight (W) and the hidden layer threshold (b) results in a unique output matrix (H) for the hidden layer. Given these conditions, the learning process of the ELM can be reformulated as a linear system.

$$\bar{\beta} = H^{-1}Y \quad (15)$$

Where H^{-1} is a generalized inverse matrix.

2.5. Data preprocessing

Prior to ingestion by the machine learning models, it is imperative to engage in preprocessing of the dataset. Within the context of this study, it is noteworthy that the original dataset is devoid of any missing values and manifests a consistent sampling interval of 30 minutes.

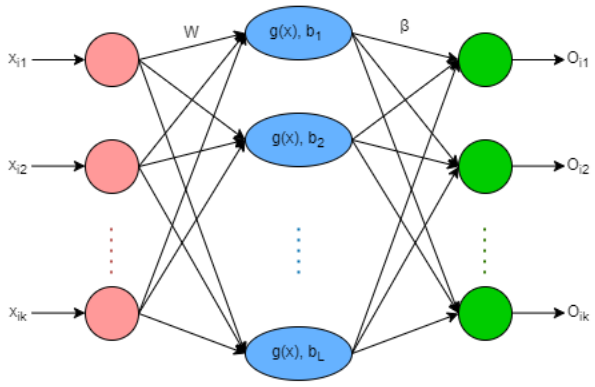

Figure 4: The structure of an ELM model

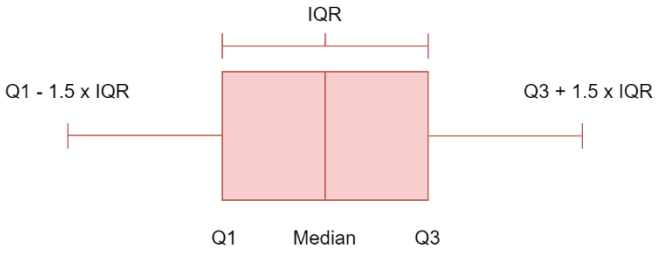

Figure 5: The Interquartile range (IQR)

Consequently, outlier detection be comes the sole preprocessing step necessary prior to deploying the dataset for forecasting purposes. The Interquartile Range (IQR) [22] is employed as the method of choice for outlier detection in this context.

The range of variations can be described using Figure 5, where:

Median: The central point that divides the dataset into upper and lower halves.

Q1, Q3: The first and third quartiles of the dataset.

Lower Bound: Q1 – 1.5 * IQR

Upper Bound: Q3 + 1.5 * IQR

The Interquartile Range (IQR) is calculated as the difference between the first quartile and the third quartile. An outlier is identified as a data point lying beyond the range defined by the Lower Bound and Upper Bound. Outliers detected are then determined to be replaced in the data set depending on their roles and importance.

2.6. Errors metrics

Error metrics are commonly used for analysis and comparison of different models. RMSE and N-RMSE are chosen as the metrics to compare the four machine learning models. RMSE and N-RMSE are defined using the following formula [23]:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(O_i - E_i)^2}{N}} \quad (16)$$

$$\text{N-RMSE} = \frac{\text{RMSE}}{\bar{O}} \times 100 \quad (17)$$

Where N was the number of validation data, O_i and E_i was the actual and estimated power value, respectively. $\bar{O}$ was the mean value of actual power value. In this study, the accuracy of the model was considered excellent when N-RMSE < 10%; good if 10% < N-RMSE < 20%; fair if 20% < N-RMSE < 30% and poor if N-RMSE ≥ 30% [23]

3. Data set

Table 1: Raw data from Nhi Ha power plant

No	Time	Output
1	6/21/2019 00:00	**0**
2	6/21/2019 00:30	**0**
3	6/21/2019 01:00	**0**
…	**…**	**…**
62159	1/08/2023 23:00	**0**
62160	1/08/2019 23:30	**0**

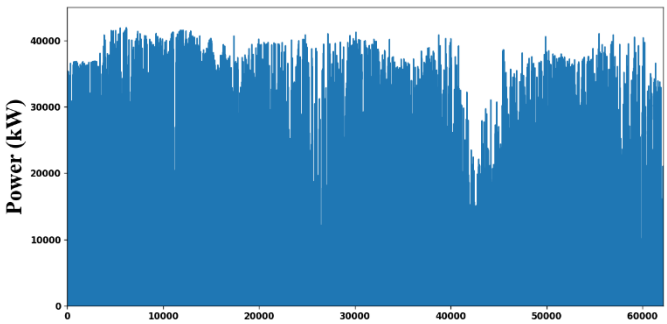

Figure 6: Output of Nhi Ha solar power plant

The data utilized in this study was meticulously gathered from the Nhi Ha Solar Power Plant, strategically situated in the Ninh Thuan province, a region in the southern part of Vietnam. The data was collected from 6/21/2019 00:00:00 to 01/08/2023 23:30:00 with a sample every 30 minutes. Because of this, there were a total of 62160 data points. Table 1 describes the raw data of the output obtained from the power plant. A visual representation of the solar power plant's output is provided in Figure 6 for further elucidation.

The output was recorded in kW. It is noteworthy to mention that the majority of deep learning models exhibit sensitivity to data scales, necessitating the standardization or normalization of the data. In the context of standardization, it is imperative that the mean of the data remains constant. Consequently, in this study, normalization of the data was executed within a range of 0 to 1.

During the application of the Interquartile Range (IQR) method for data preprocessing, various descriptive statistics were obtained. Specifically, the first quartile (Q1) was found to be 0 kW, while the third quartile (Q3) was identified as 16037.25 kW. The subsequent computation of the IQR resulted in a value of 16037.25 kW, which, in turn, guided the determination of the Upper Bound and the Lower Bound at 40093.13 kW and -24055.88 kW, respectively.

4. Results and discussion

4.1. Hyperparameter of the compared models

The specific hyperparameters of the machine learning models were presented in Table 2 and Table 3 The ANN model consists of 2 Dense layers while CNN and LSTM use only 1 Dense layer. Additionally, CNN uses 1 convolutional layer and LSTM use 2 LSTM layer.
On the other hand, ELM's hyperparameters include:

Table 2: Layer configuration for ANN, CNN and LSTM

Hyperparameter	ANN	CNN	LSTM
Conv1D	-	100	-
MaxPooling	-	2	-
LSTM1	-	-	100
LSTM2	-	-	50
Dropout	0.2	0.1	0.15
Dense1	64	50	32
Dense2	32	-	-
Activation Function	ReLU	ReLU	ReLU

Table 3: Layer configuration for ELM

Hyperparameter	ELM
Hidden neurons	100
Alpha	0.7
rbf_width	0.3
Activation Function	Sigmoid

The number of hidden neurons, alpha - the mixing coefficient for distance and dot product input activations, and rbf_width - the multiplier for the radial basis activation function.
ANN, CNN and LSTM utilize the ReLU activation function while ELM uses the Sigmoid activation function.

4.2. Forecasting result in error metrics

This section presented a comparative analysis of four machine-learning algorithms for solar power forecasting: Artificial Neural Network (ANN), Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM) and Extreme Learning Machine (ELM). The performance of these models was evaluated using two error metrics: Root Mean Squared Error (RMSE) and Normalized Root Mean Squared Error (N-RMSE). The N-RMSE was used to evaluate the overall performance of the models while RMSE was used to compare the performance between each model.

Table 4 summarized the error parameters of the machine-learning algorithms. Figures 7-12 show the predicted solar power for 1, 3, 6, 12, 24 and 48 step forecasting scenarios. Overall, using N-RMSE, it is noticeable that the performance of each model is relatively well, with excellent accuracy of all models in 1 and 3-step forecasting, and good accuracy of all models in 6, 12, 24 and 48-step forecasting. Among the 4 models evaluated, LSTM and ELM demonstrated better performance in terms of accuracy. Conversely, CNN and ANN models yielded results that were notably below expectations.

A close examination of Table 4 reveals that all four models had relatively small errors in terms of RMSE and N-RMSE. However, LSTM and ELM achieved better accuracy compared to ANN and CNN. For single-step forecasting, ELM had the lowest error values among all models, with its RMSE being 8.8 kW lower than that of LSTM. ANN and CNNperformance was found to be inferior by a small margin when compared to LSTM and ELM. The RMSE of ANN was 239.1 kW lower than the RMSE of CNN, thereby rendering the CNN model as the least effective model among the four evaluated models when considering single-step forecasting. Figure 7 illustrated the actual power and the predicted power produced by the machine-learning algorithms in single-step forecasting.

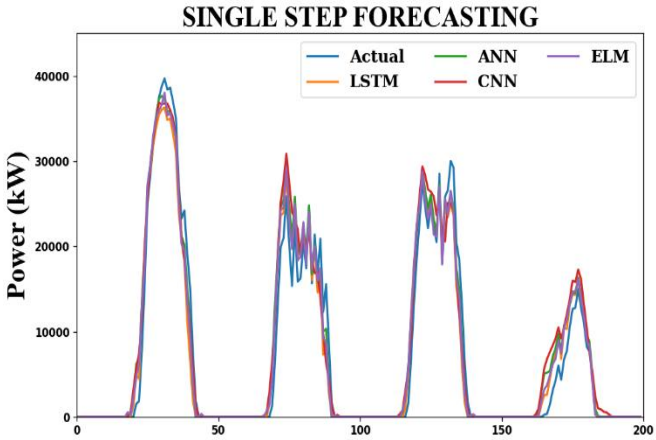

Figure 7: Actual power and predicted power of ML models in 1 step

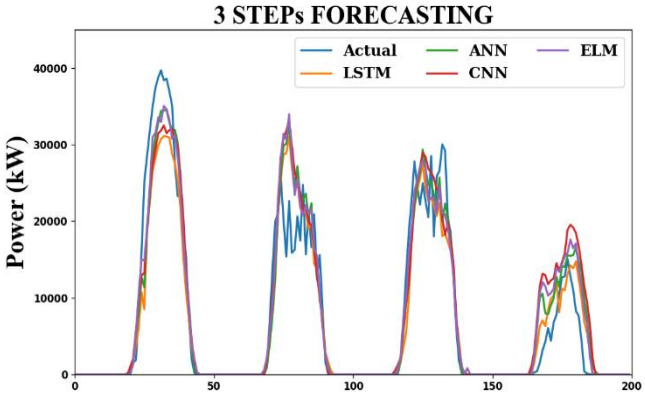

Figure 8: Actual power and predicted power of ML models in 3 steps

For multi-step forecasting with 3, 6, 12 and 24 steps ahead, the errors of all four models rose gradually, with LSTM overtook ELM in terms of accuracy, as ELM had higher RMSE than LSTM. The RMSE of LSTM were 45.6 kW, 137.6 kW, 108.9 kW and 65.1 kW lower than the RMSE of ELM for 3, 6, 12 and 24 step ahead, respectively. Despite small improvements, CNN persistently remained the least accurate model. However, it is noteworthy that the disparity in accuracy between ANN and CNN has been gradually diminishing with forecasting that has higher steps. Figure 8-11 illustrated the actual and forecasted power output using 4 machine learning methods in 3, 6, 12 and 24 steps ahead forecasting.

Finally, from 48 steps forecasting, the errors of all four models only increase by a small amount, with the ELM model surpassing the LSTM model and emerging as the most effective among the four forecasting models under consideration. The ELM model outperformed its closest competitor, LSTM, by just 7 kW in terms of RMSE. As evidenced by the data presented in Table 4, the CNN and ANN models continued to exhibit worse performance relative to the other models. The RMSE of CNN was only a 36.5 kW increase from the RMSE of ANN, making ANN the least accurate model when compared to the others model. Figure 12 provided a visual representation of the actual and predicted power output as forecasted by the different machine-learning methods in 48 steps forecasting.

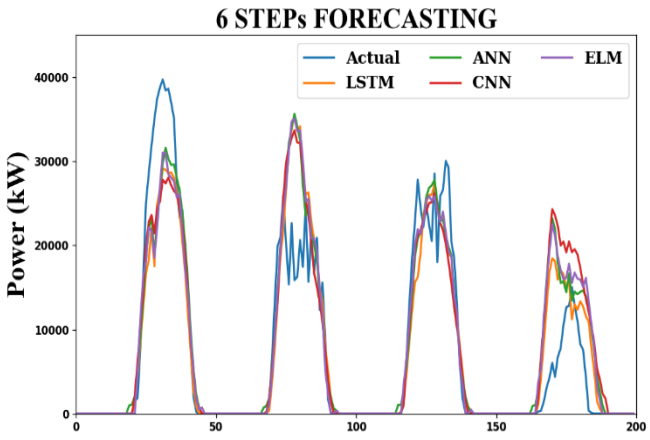

Figure 9: Actual power and predicted power of ML models in 6 steps

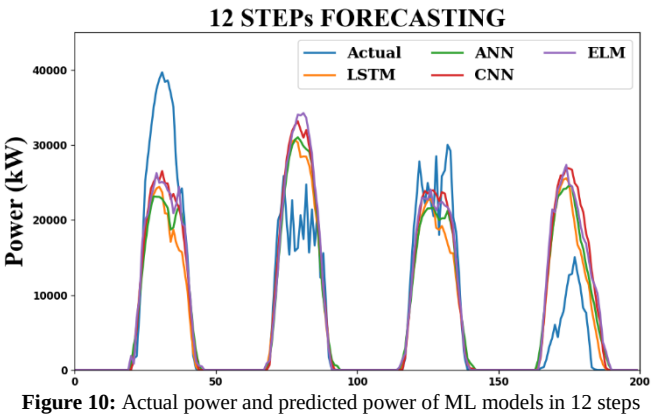

Figure 10: Actual power and predicted power of ML models in 12 steps

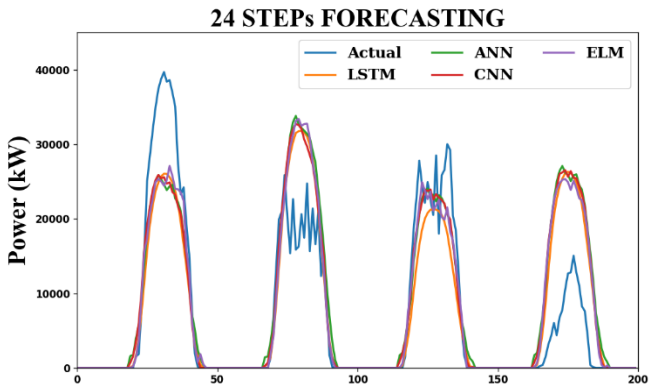

Figure 11: Actual power and predicted power of ML models in 24 steps

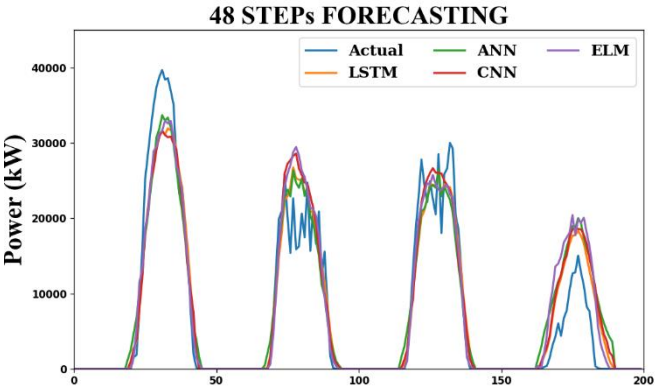

Figure 12: Actual power and predicted power of ML models in 48 steps

Based on the acquired findings, several targeted improvements can be implemented to augment the advantages for grid operators and stakeholders. In particular, the forecasting models developed for 1 and 3-step forecasting can be effectively employed to forecast solar power outputs, thereby facilitating more accurate management of solar farms. For forecasting involving higher steps, additional efforts will need to be directed toward enhancing the accuracy of the forecasting models. Consequently, grid operators will experience increased ease in efficiently administering and optimizing the power grid. Moreover, the results can assist shareholders of solar power plants in optimizing their revenues and profits by enabling them to participate in electricity markets, bid for contracts, and avoid financial losses for under or over-production of electricity.

4.3. Forecasting result in training and testing time

In addition to RMSE and n-RMSE, the training time and testing time of each model may be employed as metrics for comparative analysis. The training time denotes the duration required for model training, while the testing time represents the duration for a model to generate output.

As illustrated in Table 5, the LSTM model exhibits training and testing times of approximately 14 minutes in all step, and 5 minutes in all step, respectively.

These outcomes imply that the LSTM model demonstrates the lengthiest durations for both training and testing among the four models considered. The lengthy training period is one of the disadvantages of the LSTM model that has been mentioned in [17]. Conversely, the ELM model attains the shortest training time, reaching 0.09s at 1, 2, 4 and 48-step and 0.10s at 3,6 and 12-step. The model also has the shortest training time with the highest only being 0.78s at 48-step. The utilization of randomly generated weights in ELM serves to circumvent iterative learning processes and mitigates computational complexity. Furthermore, the training and testing times for the ANN model rank as the second lowest among the models, registering the highest recorded value of 1 minute and 26.82 seconds at 6-step, and 0.92 seconds at 6-step, respectively. Meanwhile, the CNN model follows the ANN model in terms of training and testing times, with the highest durations of 6 minutes and 57.78 seconds at 48-step, and 1.63 seconds at 24-step, respectively.

5. Conclusion.

Due to the unstable nature of solar power, forecasting of photovoltaic power output is essential for the operation of grid operators. In this paper, the accuracy of machine learning models including ANN, CNN, LSTM, and ELM was compared based on RMSE and N-RMSE errors combined with training time and testing time. All four machine learning models performed well, with minor differences in error rates. The data set had low variability, so any of the four methods yielded good results. However, applying these models to new data sets from other case studies would require readjusting and recalculating the hyperparameters, reevaluating the forecasting accuracy. To achieve better forecasting performance in future research, other techniques such as decomposition, hybrid models, or error correction should be considered.

Table 4: Evaluation of LSTM, CNN, ANN and ELM with error metrics in 1, 3, 6, 12, 24 and 48 steps

		1 step	3 steps	6 steps	12 steps	24 steps	48 steps
LSTM	**RMSE (kW)**	2671.1	3847.5	4397.0	4716.0	4781.5	4867.1
	N-RMSE (%)	6.5	9.4	10.7	11.5	11.7	11.9
CNN	**RMSE (kW)**	2930.5	4008.7	4605.0	4760.0	4851.9	4898.2
	N-RMSE (%)	7.1	9.8	11.2	11.6	11.8	11.9
ANN	**RMSE (kW)**	2698.6	3900.0	4515.3	4836.4	4894.8	4934.7
	N-RMSE (%)	6.6	9.5	11.0	11.8	11.9	12.0
ELM	**RMSE (kW)**	2662.3	3893.1	4534.6	4824.9	4846.6	4860.1
	N-RMSE (%)	6.5	9.5	11.0	11.8	11.8	11.8

Table 5: Evaluation of LSTM, CNN, ANN and ELM with training and testing time in 1, 3, 6, 12, 24 and 48 steps

		1 step	3 steps	6 steps	12 steps	24 steps	48 steps
LSTM	**Testing time**	00:05.24	00:04.90	00:05.86	00:05.19	00:05.19	00:04.81
	Training time	15:07.81	14:37.53	13:10.21	13:48.68	14:27.20	13:24.04
CNN	**Testing time**	00:01.40	00:00.79	00:01.12	00:00.68	00:01.63	00:01.19
	Training time	02:40.29	04:56.81	04:36.76	04:14.77	04:23.62	06:57.78
ANN	**Testing time**	00:00.64	00:00.58	00:00.92	00:00.60	00:00.52	00:00.67
	Training time	00:45.68	01:26.01	01:26.82	01:16.67	01:08.64	01:17.31
ELM	**Testing time**	00:00.09	00:00.10	00:00.10	00:00.10	00:00.09	00:00.09
	Training time	00:00.70	00:00.72	00:00.79	00:00.79	00:00.75	00:00.78

References

[1] Prime Minister of Vietnam, "Decision on mechanisms to promote the development of solar power projects in Viet Nam," Apr. 2020.

[2] P. Singla, M. Duhan, and S. Saroha, "A comprehensive review and analysis of solar forecasting techniques," *Front. Energy*, vol. 16, no. 2, pp. 187–223, Apr. 2022, doi: 10.1007/s11708-021-0722-7.

[3] D. C. Montgomery, C. L. Jennings, and M. Kulahci, *Introduction to time series analysis and forecasting*. John Wiley & Sons, 2015.

[4] K. Benmouiza and A. Cheknane, "Small-scale solar radiation forecasting using ARMA and nonlinear autoregressive neural network models," *Theor Appl Climatol*, vol. 124, no. 3–4, pp. 945–958, May 2016, doi: 10.1007/s00704-015-1469-z.

[5] J. Lu *et al.*, "Two-Tier Reactive Power and Voltage Control Strategy Based on ARMA Renewable Power Forecasting Models," *Energies*, vol. 10, no. 10, p. 1518, Oct. 2017, doi: 10.3390/en10101518.

[6] W. Wang, K. Chau, D. Xu, and X.-Y. Chen, "Improving Forecasting Accuracy of Annual Runoff Time Series Using ARIMA Based on EEMD Decomposition," *Water Resour Manage*, vol. 29, no. 8, pp. 2655–2675, Jun. 2015, doi: 10.1007/s11269-015-0962-6.

[7] M. Q. Raza, M. Nadarajah, and C. Ekanayake, "On recent advances in PV output power forecast," *Solar Energy*, vol. 136, pp. 125–144, Oct. 2016, doi: 10.1016/j.solener.2016.06.073.

[8] N. N. V. Nhat, D. N. Huu, and T. N. T. Hoai, "Evaluating the EEMD-LSTM model for short-term forecasting of industrial power load: A case study in Vietnam," *Int. J. Renew. Energy Dev.*, vol. 12, no. 5, pp. 881–890, Sep. 2023, doi: 10.14710/ijred.2023.55078.

[9] N. T. H. Thu, P. N. Van, and P. Q. Bao, "Multi-step Ahead Wind Speed Forecasting Based on a Bi-LSTM Network Combined with Decomposition Technique," in *Computational Intelligence Methods for Green Technology and Sustainable Development*, vol. 567, Y.-P. Huang, W.-J. Wang, H. A. Quoc, H.-G. Le, and H.-N. Quach, Eds., in Lecture Notes in Networks and Systems, vol. 567. , Cham: Springer International Publishing, 2023, pp. 569–580. doi: 10.1007/978-3-031-19694-2_50.

[10] Y. Huang, "Advances in Artificial Neural Networks – Methodological Development and Application," *Algorithms*, vol. 2, no. 3, pp. 973–1007, Aug. 2009, doi: 10.3390/algor2030973.

[11] M. M. Mijwil, "Artificial Neural Networks Advantages and Disadvantages," *Mesopotamian Journal of Big Data*, vol. 2021, pp. 29–31, Aug. 2021, doi: 10.58496/MJBD/2021/006.

[12] C. Wang and Z. Peng, "Design and Implementation of an Object Detection System Using Faster R-CNN," in *2019 International Conference on Robots & Intelligent System (ICRIS)*, Haikou, China: IEEE, Jun. 2019, pp. 204–206. doi: 10.1109/ICRIS.2019.00060.

[13] N. Jmour, S. Zayen, and A. Abdelkrim, "Convolutional neural networks for image classification," in *2018 International Conference on Advanced Systems and Electric Technologies (IC_ASET)*, Hammamet: IEEE, Mar. 2018, pp. 397–402. doi: 10.1109/ASET.2018.8379889.

[14] D. Wang, H. Yu, D. Wang, and G. Li, "Face Recognition System Based on CNN," in *2020 International Conference on Computer Information and Big Data Applications (CIBDA)*, Guiyang, China: IEEE, Apr. 2020, pp. 470–473. doi: 10.1109/CIBDA50819.2020.00111.

[15] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *J Big Data*, vol. 8, no. 1, p. 53, Mar. 2021, doi: 10.1186/s40537-021-00444-8.

[16] H. Sak, A. Senior, and F. Beaufays, "Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition," 2014, doi: 10.48550/ARXIV.1402.1128.

[17] G. Van Houdt, C. Mosquera, and G. Nápoles, "A review on the long short-term memory model," *Artif Intell Rev*, vol. 53, no. 8, pp. 5929–5955, Dec. 2020, doi: 10.1007/s10462-020-09838-1.

[18] A. K. Poddar and Dr. R. Rani, "Hybrid Architecture using CNN and LSTM for Image Captioning in Hindi Language," *Procedia Computer Science*, vol. 218, pp. 686–696, 2023, doi: 10.1016/j.procs.2023.01.049.

[19] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[20] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, Dec. 2006, doi: 10.1016/j.neucom.2005.12.126.

[21] F. Mercaldo, L. Brunese, F. Martinelli, A. Santone, and M. Cesarelli, "Experimenting with Extreme Learning Machine for Biomedical Image Classification," *Applied Sciences*, vol. 13, no. 14, p. 8558, Jul. 2023, doi: 10.3390/app13148558.

[22] N. C. Schwertman, M. A. Owens, and R. Adnan, "A simple more general boxplot method for identifying outliers," *Computational Statistics & Data Analysis*, vol. 47, no. 1, pp. 165–174, Aug. 2004, doi: 10.1016/j.csda.2003.10.012.

[23] M.-F. Li, X.-P. Tang, W. Wu, and H.-B. Liu, "General models for estimating daily global solar radiation for different solar radiation zones in mainland China," *Energy Conversion and Management*, vol. 70, pp. 139–148, Jun. 2013, doi: 10.1016/j.enconman.2013.03.004.